

# Android Studio

O Android Studio é uma IDE oficial do Google para o desenvolvimento de aplicativos Android. Com base no editor de código e nas ferramentas para desenvolvedores avançados do [IntelliJ IDEA](#), oferecendo um conjunto de ferramentas poderosas e recursos especializados. O Android Studio oferece ainda mais recursos para aumentar a produtividade ao criar apps Android, como:

Ambiente unificado que possibilita o desenvolvimento para todos dispositivos **Android**;

- Edição em tempo real para atualizar elementos combináveis em emuladores e dispositivos físicos;
- Frameworks e ferramentas de teste cheios de possibilidades;
- Compatibilidade integrada com o [Google Cloud Platform](#), facilitando a integração do Google Cloud Messaging e do App Engine;
- Modelos de código e integração com GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código;

## Principais funcionalidades

**Design e Layout** - O Android Studio inclui um designer gráfico que permite a criação de interfaces de usuário. Oferecendo uma visualização em tempo real do layout, permitindo que você arraste e solte componentes da interface e ajuste suas propriedades para personalizar a aparência do aplicativo.

**Gerenciador de Dependências** - O Android Studio facilita o gerenciamento das dependências do seu projeto usando o Gradle. O Gradle é um sistema de construção poderoso que permite adicionar bibliotecas e componentes externos ao seu aplicativo, simplificando a integração de recursos adicionais.

**Depuração e Testes** - O Android Studio fornece recursos avançados para depuração e testes, oferecendo depuração passo a passo, visualização de pilha de chamadas, inspeção de variáveis, breakpoints e muito mais. Além disso, o usuário pode executar testes automatizados para garantir a qualidade e o desempenho do seu aplicativo.

**Emulador e Dispositivos Virtuais** - O Android Studio fornece recursos avançados para depuração e testes, oferecendo depuração passo a passo, visualização de pilha de chamadas, inspeção de variáveis, breakpoints e muito mais. Além disso, o usuário pode executar testes automatizados para garantir a qualidade e o desempenho do seu aplicativo.

**Compatibilidade Google** - O Android Studio por ser um IDE desenvolvida pelo Google, traz consigo um leque de recursos que podem ser facilmente manipulados para trazer mais produtividade para o time de desenvolvimento que opte por trabalhar com ele. Alguns exemplos de serviços que são facilmente manipulados são : [Google Play Games](#), [Firebase](#), [Google Cloud](#), [entre outros](#).

# Linguagens são mais apropriadas

O Android Studio é mais amplamente utilizado para desenvolvimento nativo, que são escritos usando principalmente a linguagem **Java** e, mais recentemente, a linguagem **Kotlin**. Além dessas linguagens, o Android Studio também oferece suporte a outras linguagens, como **C++** e **C**, para desenvolvimento de componentes nativos de baixo nível, como bibliotecas ou módulos específicos.

## Java no Android Studio

**Java** é uma linguagem orientada a objetos, o que permite uma estrutura clara e modular para o desenvolvimento de aplicações. Ajudando na manutenção do código, reutilização de componentes e facilidade no trabalho em equipe. Com isso, vem sendo amplamente utilizada no desenvolvimento de aplicativos móveis. Isso significa que traz consigo uma variedade de bibliotecas, documentação e recursos disponíveis. Dito isso, usar o **Java** no Android Studio permite a compatibilidade com esses aplicativos existentes e facilita a migração gradual para o **Kotlin**, se desejado.

## Kotlin no Android Studio

**Kotlin** traz consigo uma sintaxe concisa e expressiva, o que reduz a quantidade de código. Isso aumenta a produtividade e torna o código mais legível e fácil de entender. Sendo inteiramente compatível com o **Java**, sendo possível combinar código **Java** e **Kotlin** no mesmo projeto. Facilitando a utilização de Kotlin em projetos que são inicialmente **Java**.

## C++ no Android Studio

Se o usuário já possui bibliotecas em **C++** que deseja usar, o **Android Studio** permite uma integração efetiva ao projeto usando o **Android NDK (Native Development Kit)** e aproveitar a reutilização de código existente. Além disso, o **C++** pode acessar APIs nativas do Android que não estão disponíveis diretamente no **Java** ou **Kotlin**. Permitindo a integração com bibliotecas de terceiros ou a implementação de funcionalidades específicas do sistema operacional.

## C no Android Studio

Se o usuário possui código legado em **C** que deseja portar para o Android, é possível integrá-lo ao projeto e aproveitar esse código existente, por meio do **Android NDK**. Além disso, o **Android NDK** oferece suporte ao desenvolvimento de componentes de baixo nível em **C**, como kernel do sistema operacional, drivers de dispositivos, módulos do sistema, entre outros.

# Ferramenta Git no Android Studio

O Android Studio possui uma integração nativa com o Git, permitindo que os desenvolvedores acessem todas as funcionalidades de forma simples e efetiva, trazendo uma interface gráfica para priorizar a produtividade da equipe de desenvolvimento.

A IDE também oferece suporte a recursos avançados, como resolução de conflitos, visualização de diferenças entre branches e histórico de alterações mais detalhado. A integração nativa do Git no Android Studio torna o controle de versão e a colaboração em projetos Android mais eficientes e convenientes para os desenvolvedores.